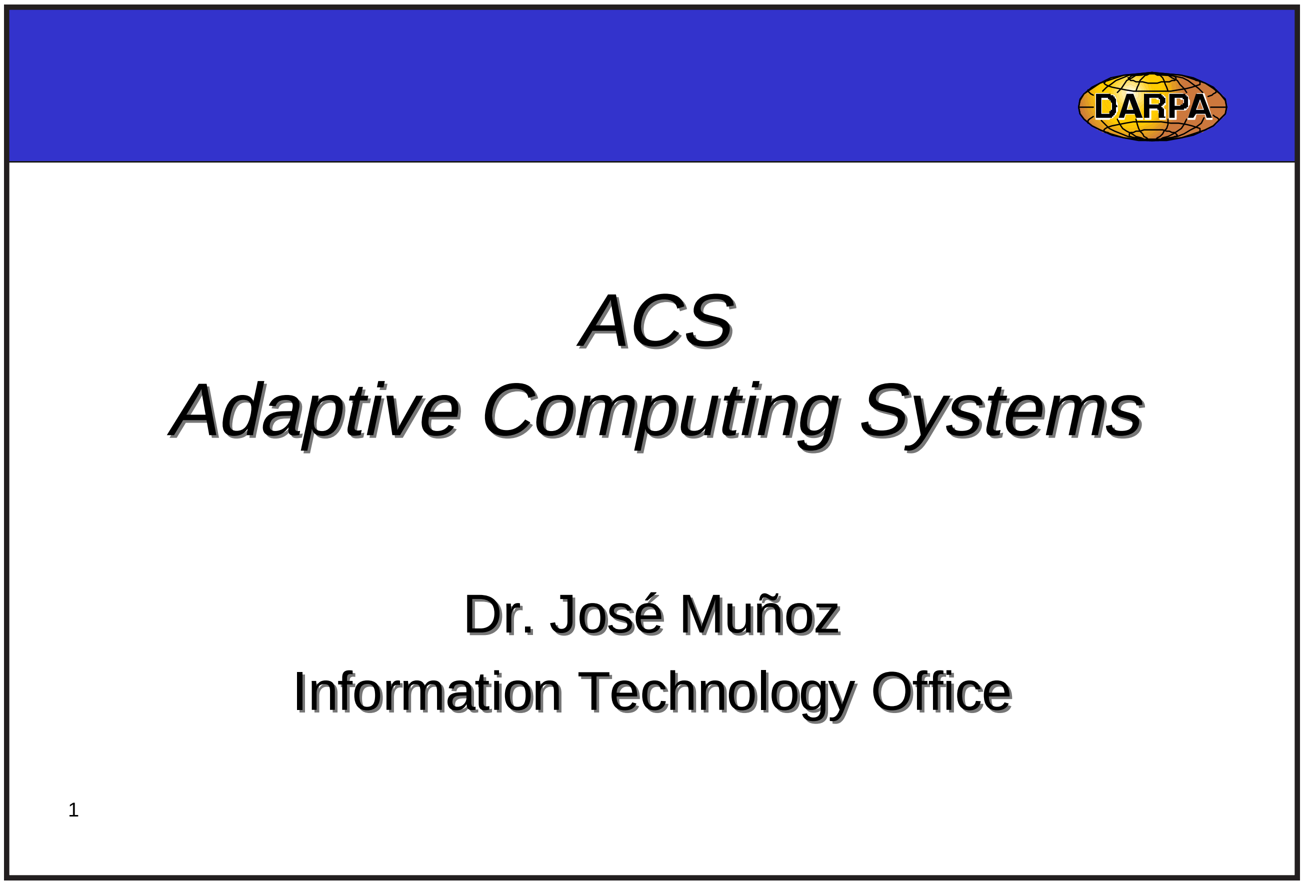


ACS
Adaptive Computing Systems

Dr. José Muñoz

Information Technology Office

1

ACS will turn around, *revolutionize*, the way we currently think about hardware and software and will provide capabilities desperately needed to address the threats and rapidly evolving mission requirements of the Department of Defense.

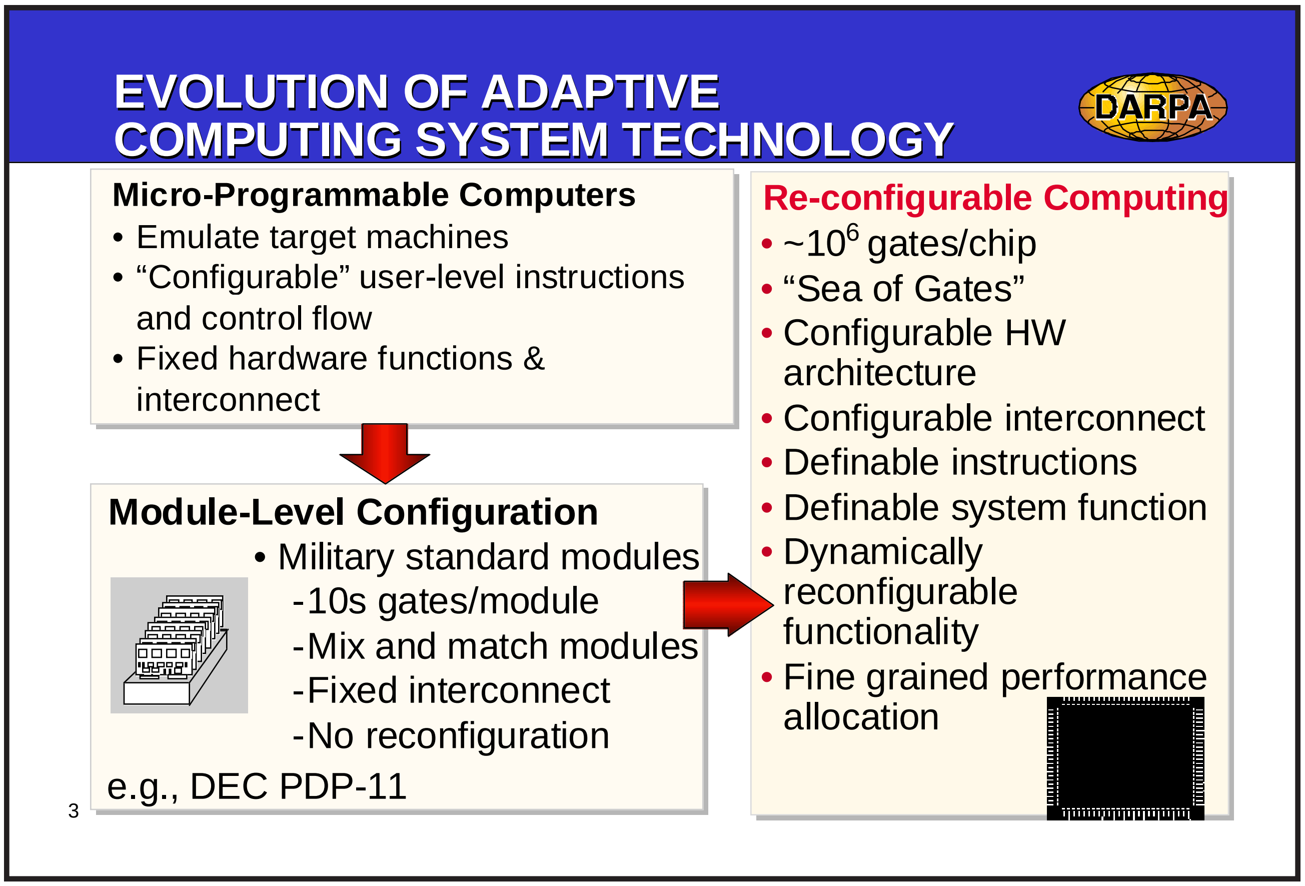
ADAPTIVE COMPUTING SYSTEMS

- Dynamic adaptation to threats
- Extended mission capabilities
- Seamless, complete, life cycle performance upgrades
- Commodity technology easily augmented

Adaptive Computing and Communications Technology for Evolving Defense Systems

The military in 1975 consumed 17% of the total semiconductor market based on dollars. In 1995 that figure was less than 1%. As a result, the military is in a position of reduced influence, but has applications that are extremely stressing and make significant demands on computational and communication resources. Further complicating the picture is that it is projected that by the year 2000 about 14 semiconductor-based devices will become obsolescent, and hence not available to the military, per day!

When completed, the Adaptive Computing Systems (ACS) program will provide to the warfighter a technology that can adapt to changes in algorithms, data, and mission stages. Regarding security issues, ACS-based systems will respond in innovative ways to fault tolerance and security concerns while at the same time providing performance benefits orders of magnitude above conventional approaches.



EVOLUTION OF ADAPTIVE COMPUTING SYSTEM TECHNOLOGY

DARPA

Micro-Programmable Computers

- Emulate target machines
- "Configurable" user-level instructions and control flow
- Fixed hardware functions & interconnect

Module-Level Configuration

- Military standard modules
  - 10s gates/module
  - Mix and match modules
  - Fixed interconnect
  - No reconfiguration

e.g., DEC PDP-11

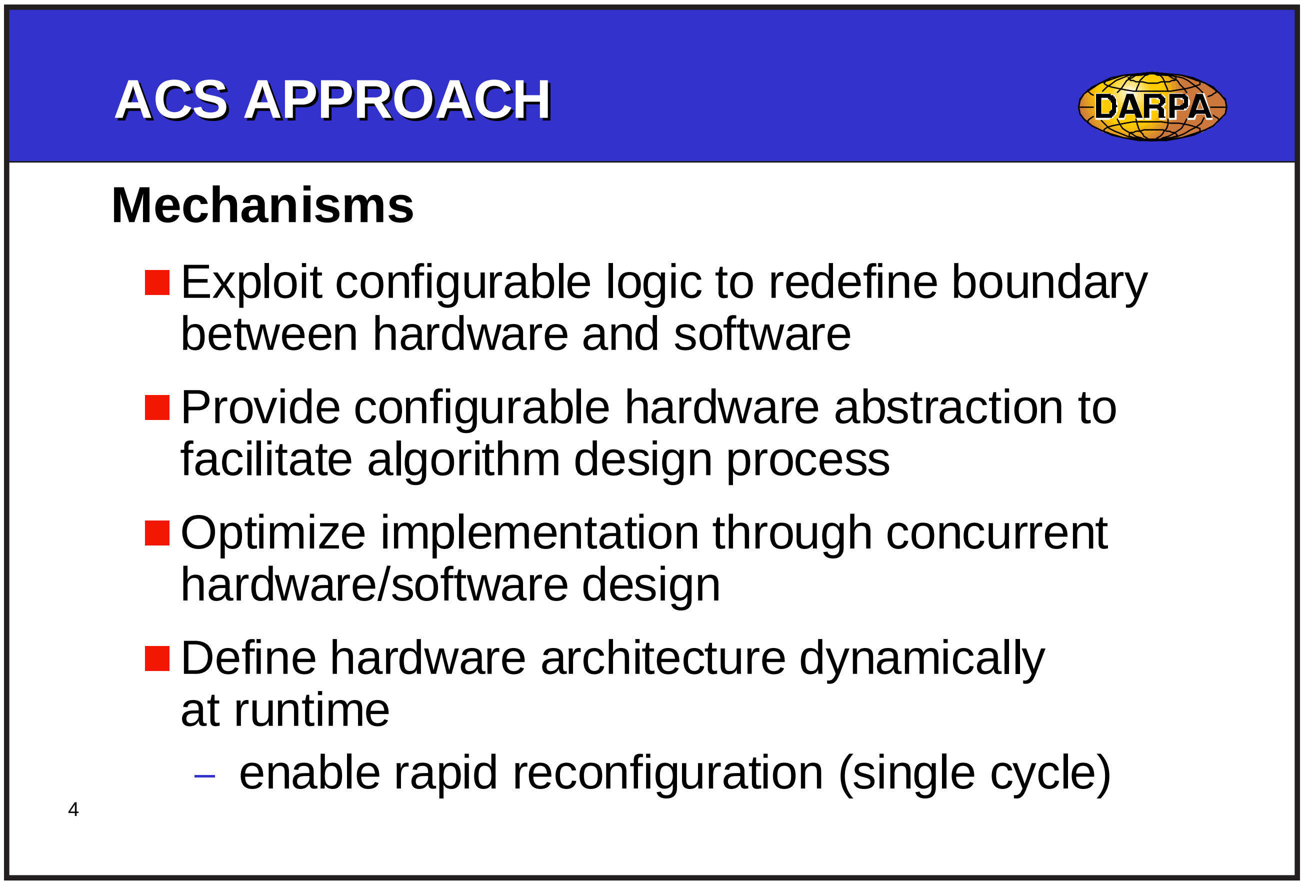
Re-configurable Computing

- ~10^6 gates/chip
- "Sea of Gates"
- Configurable HW architecture
- Configurable interconnect
- Definable instructions
- Definable system function
- Dynamically reconfigurable functionality
- Fine grained performance allocation

3

Let's take a brief look at the evolution of "adaptable" or configurable computing. Back in the '70s we had micro-programmable computers. These enabled a user to create his own instructions and control flow and were used primarily to emulate other machines. They did not provide any additional hardware functionality, and you were restricted to use the interconnect and datapaths provided by the vendor. You also had the ability to "adapt" at the module level using relatively low-complexity hardware modules. "Configuration" here consisted of the mixing and matching of boards to create a system tailored to a specific set of requirements. Again, you had to live within the fixed interconnect provided, and there certainly was no capability such as dynamic reconfiguration.

ACS will provide technologies enabling you to create, "on-the-fly" if required, the computing and interconnect resources required to meet a set of requirements. It will result in a "sea-of-gates" with on the order of a million gates per chip, that can be used to define not only instructions but also how these instructions are interconnected, or "wired." Utilizing such an approach will enable the creation of systems specifically tailored to meet requirements needs, down to the chip level. You utilize only the resources demanded by the application and only for so long as they are required…computing by the micron2/sec.



ACS APPROACH

DARPA

Mechanisms

- Exploit configurable logic to redefine boundary between hardware and software
- Provide configurable hardware abstraction to facilitate algorithm design process
- Optimize implementation through concurrent hardware/software design
- Define hardware architecture dynamically at runtime
  - enable rapid reconfiguration (single cycle)

4

The ability to create new hardware, at runtime, using software mechanisms, will result in a true paradigm shift. The boundary that currently exists between hardware and software will be removed as the user will not have to settle for the architecture provided by a vendor; but instead can specify and create algorithm specific circuits in response to application specific requirements.

The implications of such a technology are wide and far reaching. Imagine you're an algorithm designer and are now given the capability of specifying the number of bits of precision required for your problem? ACS will develop the COTS based hardware components, programming, runtime support environment and application development environments to enable the user to "reach through" the hardware layer, at runtime, and create the computing resources tailored to meet his specific requirements.

The key to all of these benefits is to provide them in a manner that is transparent and "natural" to the user.

COMPLEX SYSTEM HARDWARE EMULATION: TECHNOLOGY BASELINE

State of the Art

Sponsor: DARPA Microsystems Pgm
Performer: Virtual Machine Works
Project: Virtual Wires

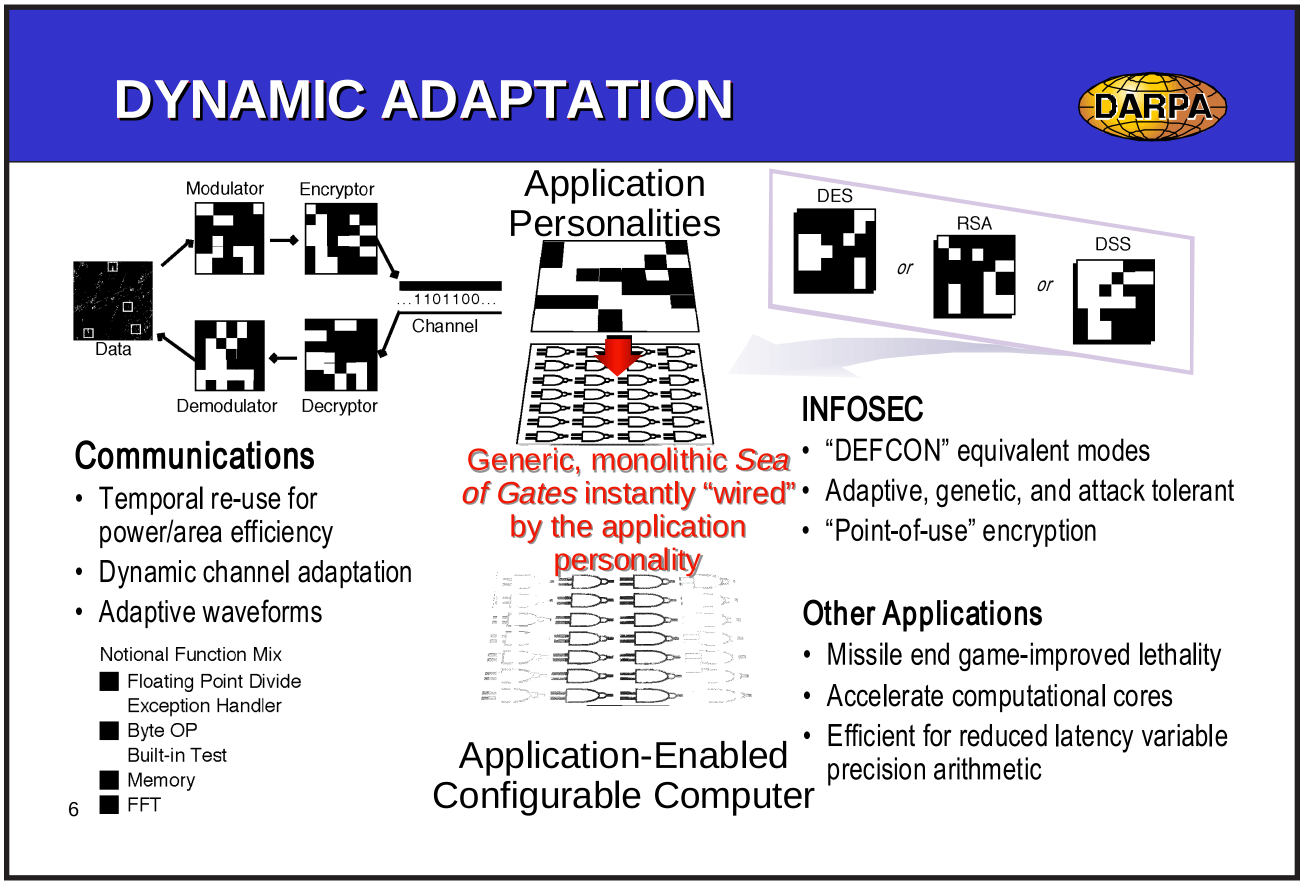
- 225K Emulation Gates
- 0.5 MHz Emulation Rate
- 6-8X Cost Reduction
- 4 Mbits Modular Memory

Pentium-Pro Class Emulation
- 1M Total Gates (4-6 boards)
- Today's technology would take "only" one board

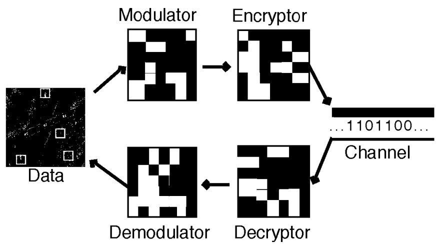
5

Shown here is the "sea-of-gates" using circa '95 technology. It's "pizza box" size, containing 64 field programmable gate arrays, FPGA, and contains about 225,000 gates running at 1/2 megahertz and has about 1/2 Mbyte of memory. It could be used to emulate a Pentium-Pro class architecture, requiring about one million gates and therefore 4-6 boards of this size, resulting in a cost savings of 6-8X over other approaches that might be used to study the behavior of architectures of this class. Using today's technology, i.e., 1997, we could do the same job with a single board. By the end of this program, we'll be able to achieve this on a single chip!

FPGAs have been used very much in this fashion, i.e., to emulate microprocessors or application specific integrated circuits (ASICs). ACS will move the FPGA technology, and hence its optimization point, further out, leveraging dynamic configuration to achieve increased performance.



DYNAMIC ADAPTATION



Communications

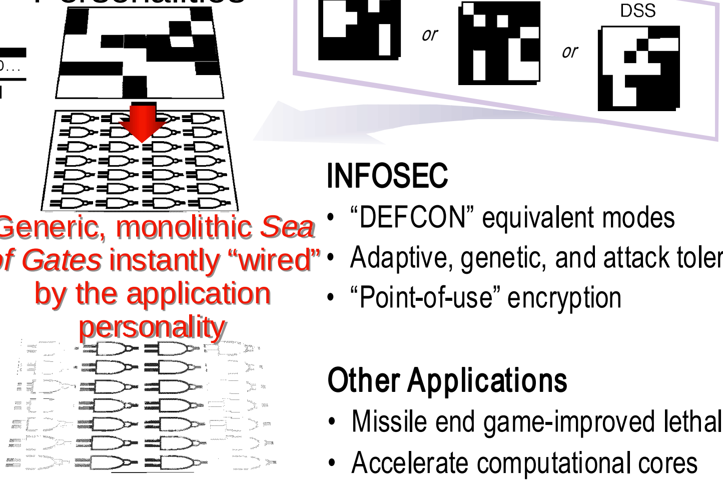
- Temporal re-use for power/area efficiency
- Dynamic channel adaptation
- Adaptive waveforms

Notional Function Mix

- ■ Floating Point Divide
- Exception Handler
- ■ Byte OP
- Built-in Test
- ■ Memory
- ■ FFT

6

Application Personalities



Generic, monolithic *Sea of Gates* instantly "wired" by the application personality

Application-Enabled Configurable Computer

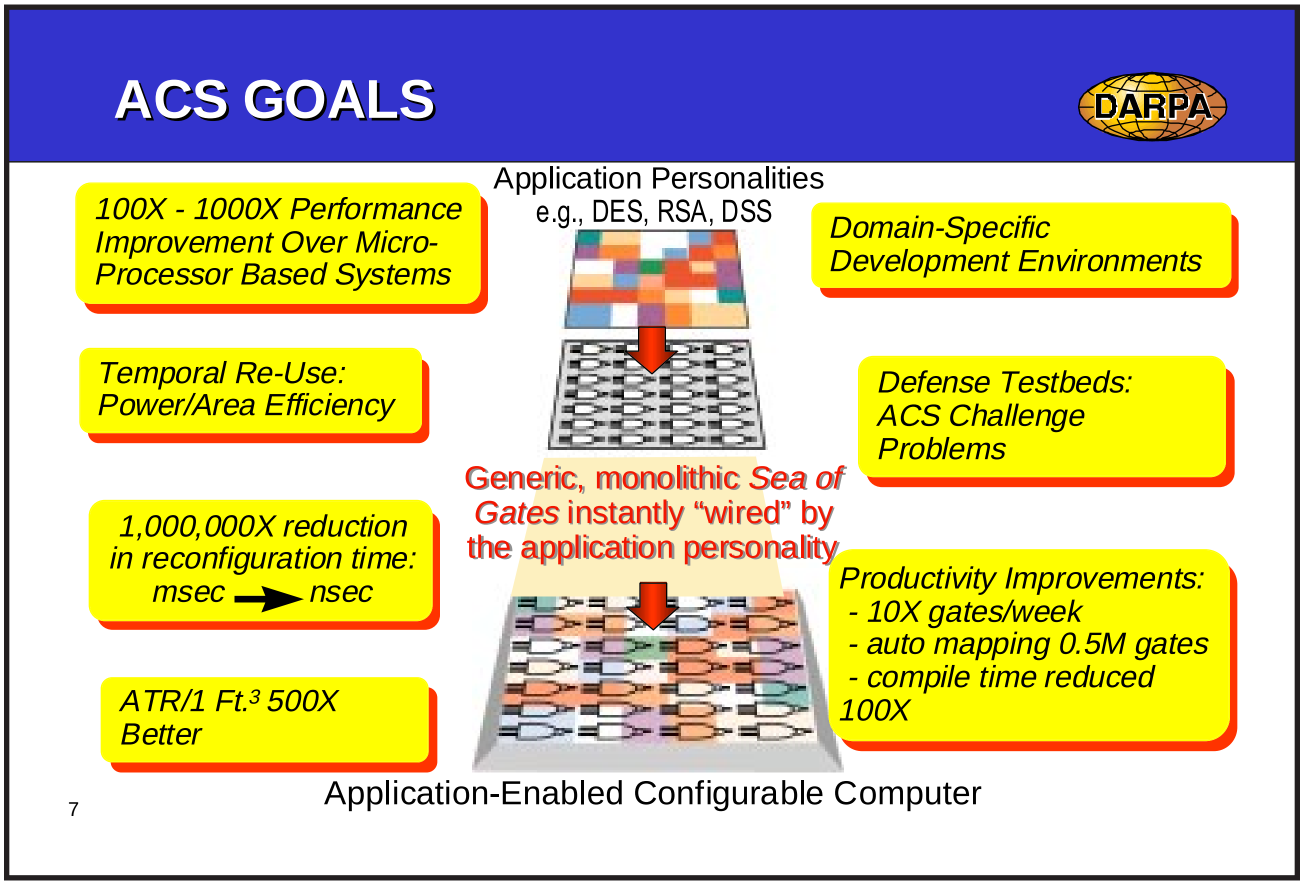
INFOSEC

- "DEFCON" equivalent modes
- Adaptive, genetic, and attack tolerant
- "Point-of-use" encryption

Other Applications

- Missile end game-improved lethality
- Accelerate computational cores
- Efficient for reduced latency variable precision arithmetic

We've seen why this technology is important. How and where might this adaptable computing systems technology be used? Adaptation will become an increasingly important attribute of future Defense systems. Various phases of a computation could be swapped in and out of a configurable system, as suggested in the above modulator/demodulator example. Similarly, we may dynamically adopt a new crypto algorithm, change our model of computation in response to different mission stages, dynamically adapt to changes in channel conditions, or exploit changes in the dynamic range of a computation using variable precision arithmetic. The key thing here is the need to respond rapidly and to be flexible.



ACS GOALS

DARPA

Application Personalities
e.g., DES, RSA, DSS

100X - 1000X Performance Improvement Over Micro-Processor Based Systems

Domain-Specific Development Environments

Temporal Re-Use: Power/Area Efficiency

Defense Testbeds: ACS Challenge Problems

Generic, monolithic *Sea of Gates* instantly "wired" by the application personality

1,000,000X reduction in reconfiguration time: msec → nsec

Productivity Improvements:
- 10X gates/week
- auto mapping 0.5M gates
- compile time reduced 100X

ATR/1 Ft.³ 500X Better

Application-Enabled Configurable Computer

7

Performance

- Continuous, sustained 100x-1000x performance improvement on critical defense signal processing applications

Cost

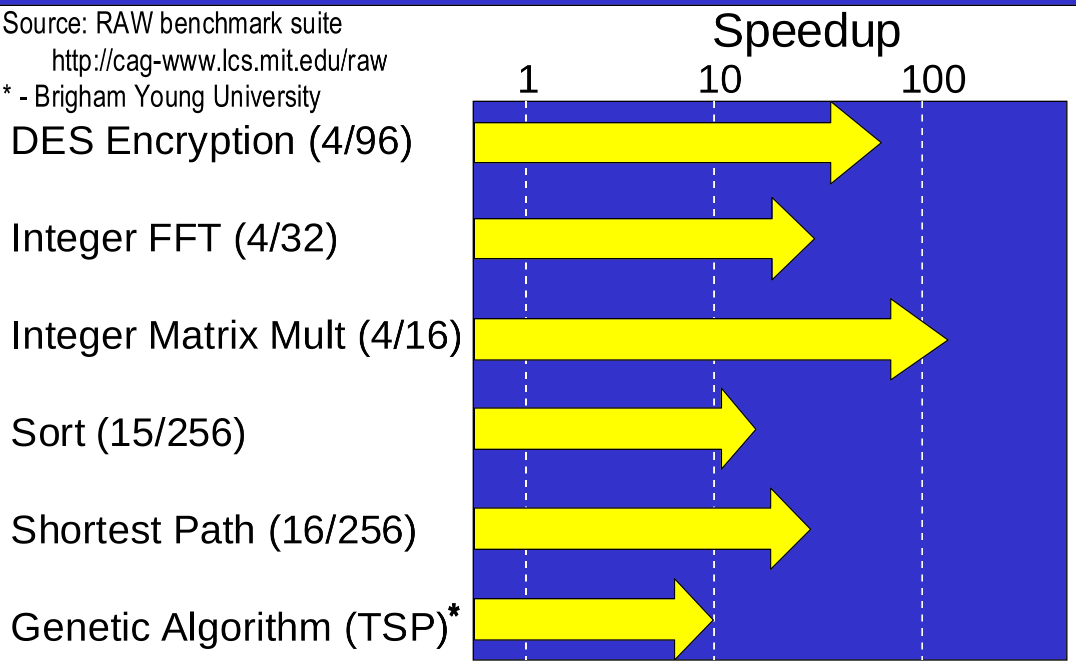
- Cost-effective Defense ACS components…replacement for ASICs
- Reduced life-cycle costs, simplified logistics…commodity components

Efficiency

- Direct manipulation of hardware architecture

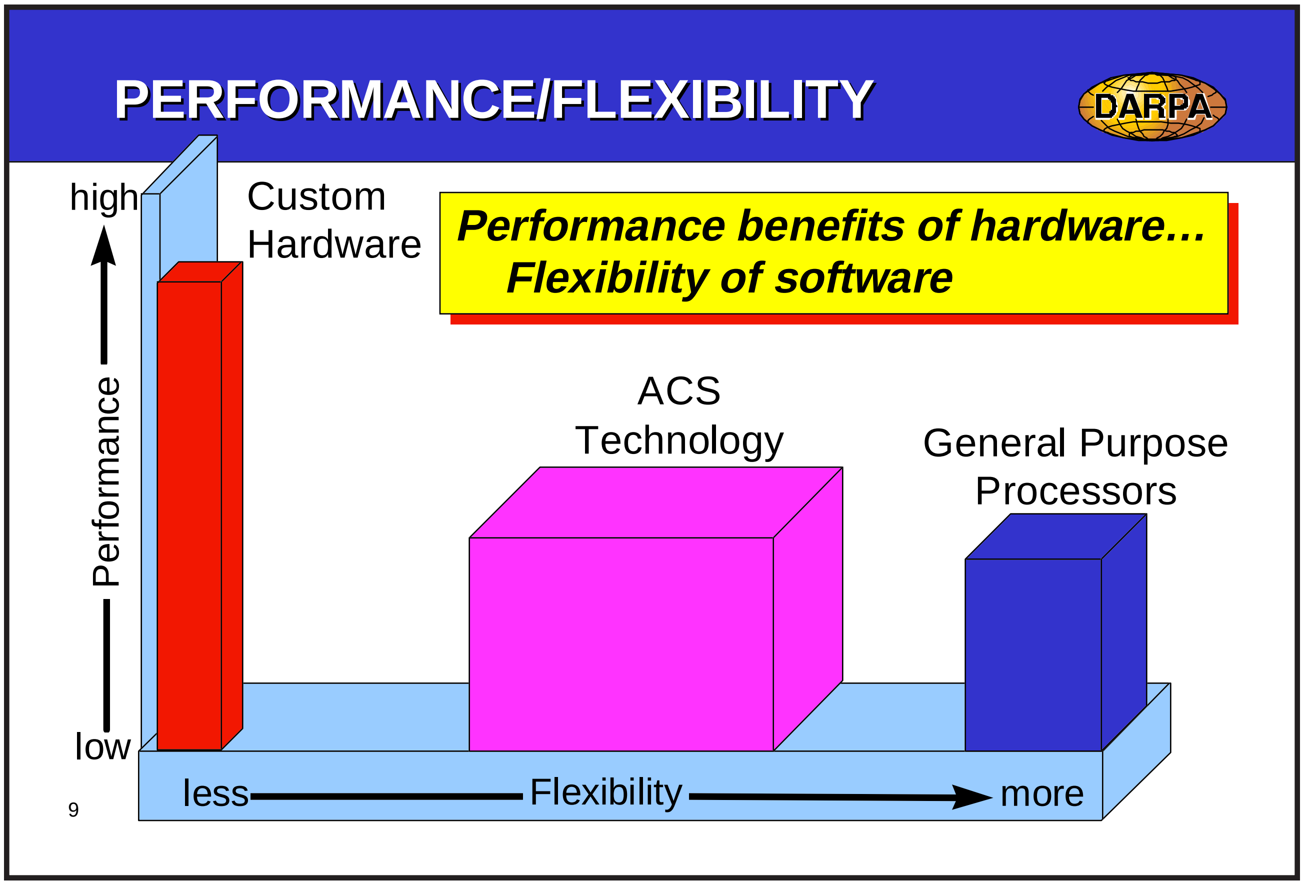
MOTIVATION

Source: RAW benchmark suite
http://cag-www.lcs.mit.edu/raw
* - Brigham Young University



Shown here are some very conservative speedups, achieved by scientists from MIT and BYU, using this technology on a few interesting applications. There have been many instances, over the past 10 years, with speedups even greater than shown here. While these results have been impressive and demonstrative of significant accomplishment…they typically require heroic efforts. However, they do provide compelling evidence that this technology can be exploited to perform computational tasks.

The reason heroic efforts are required is the lack of tool support and the significantly long compilation times required to develop these applications. Often, the user is required not only to be a domain expert but a hardware expert as well and must be able to formulate his problem using mechanisms that are familiar only to hardware developers. This, coupled with issues such as extremely long compilation times (hours as opposed to seconds), reduces productivity.

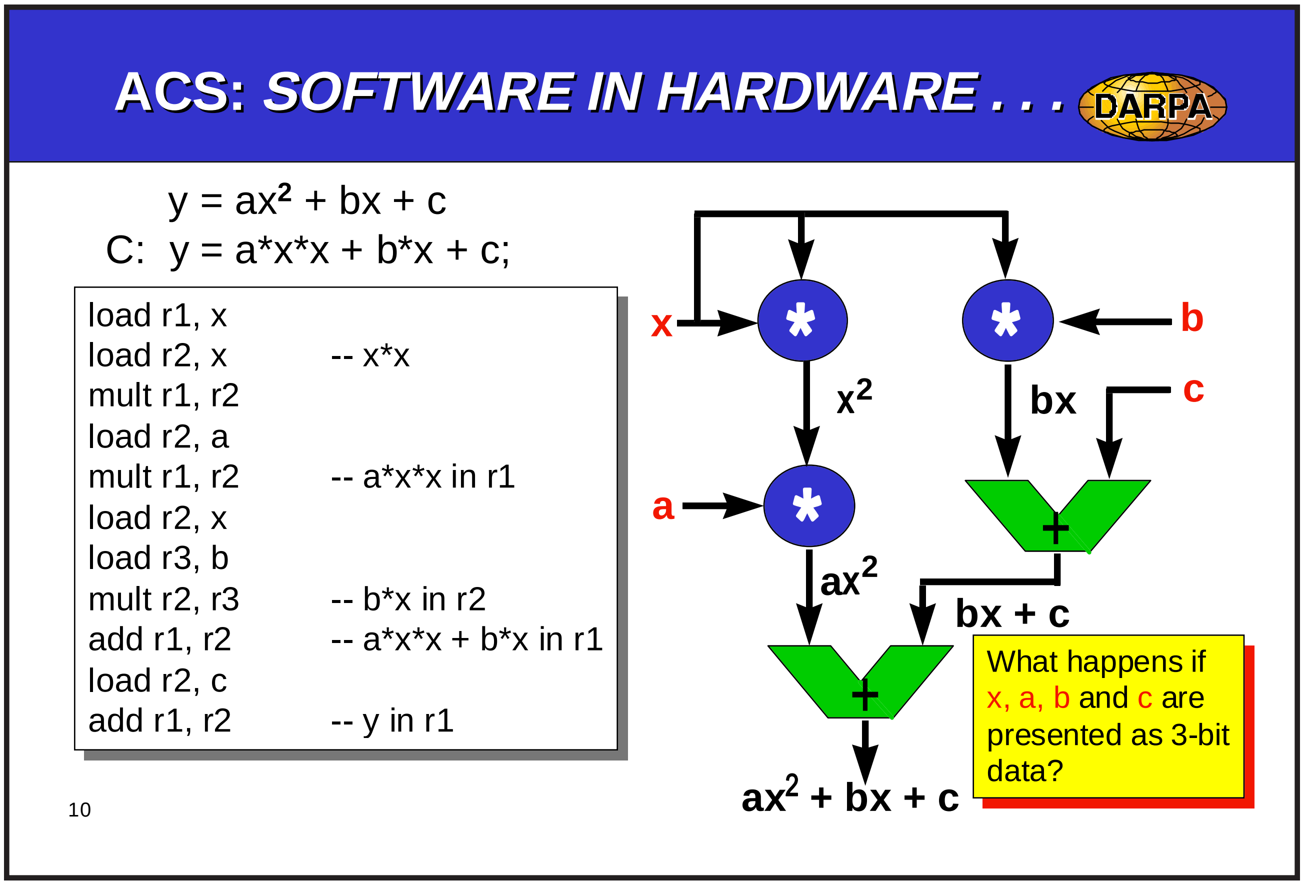


PERFORMANCE/FLEXIBILITY

Performance benefits of hardware…
Flexibility of software

Shown here is the computing space…with performance on the vertical axis and flexibility on the horizontal. Custom hardware, such as ASICs, provides significant performance advantages at the expense of flexibility. Typically, you cannot use an ASIC for processing outside its designed parameters. It achieves this performance at the cost of reduced flexibility. General purpose processors provide a great deal of flexibility… they are able to do word processing, solve complex mathematical operations or play the latest video game. These are things that were not necessarily envisioned when the processor was developed.

The technology being presented in ACS fits very nicely in the middle. It can provide the performance benefits of hardware and the flexibility of software… because you create the hardware using technologies and environments adapted from the software domain. You can very much think of this as “agile” or “malleable” hardware, because it’s hardware that you can replace on-the-fly in response to changing demands.



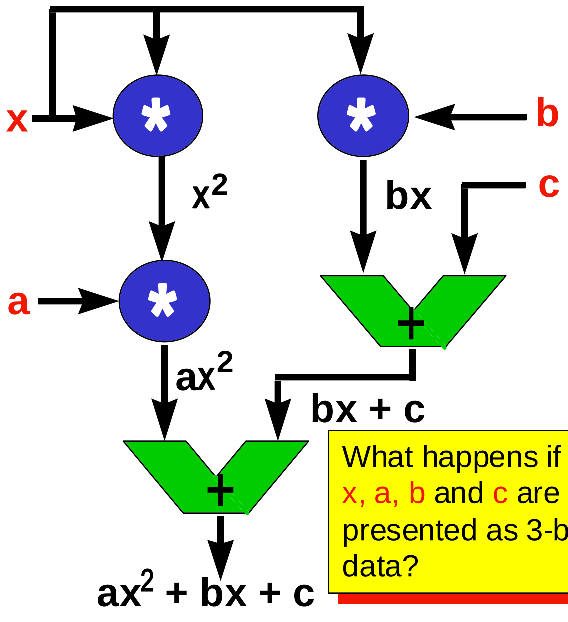
ACS: *SOFTWARE IN HARDWARE . . .*

DARPA

$y = ax^2 + bx + c$

C: `y = a*x*x + b*x + c;`

```
load r1, x
load r2, x          -- x*x
mult r1, r2
load r2, a
mult r1, r2         -- a*x*x in r1
load r2, x
load r3, b
mult r2, r3         -- b*x in r2
add r1, r2          -- a*x*x + b*x in r1
load r2, c
add r1, r2          -- y in r1
```



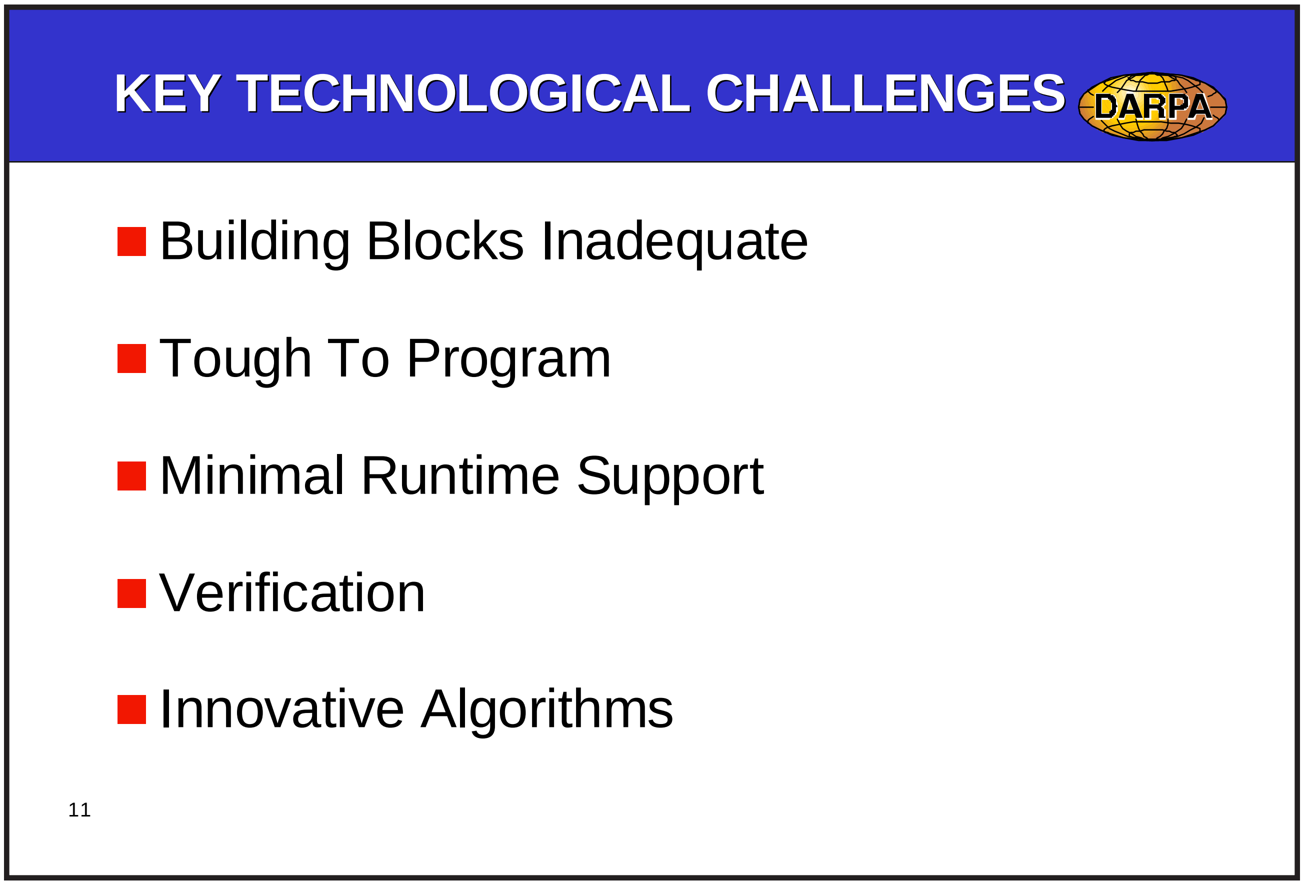
What happens if x, a, b and c are presented as 3-bit data?

10

What do we mean by “software in hardware”? Shown is the binomial equation ax**2 + bx + c, written first as a mathematician might express it, followed by a representation as it might be expressed using the popular C programming language. The box on the left shows what a typical compiler might generate from the provided C code using a generic assembler language representation. The result is 11 steps of assembly code.

On the right is shown a circuit which might be created using the same input C representation. The circles are multipliers, the V-shaped objects are adders and pathways are presented using arrows with the inputs to the equation (x, a, b and c) shown in red. Using this circuit we could arrive at a solution in 4 steps. So even on such a trivial example such as this we can have speed-ups of 3-4 over a conventional processor.

The benefits of this technology are actually demonstrated if we consider that the data is actually being presented in a 3-bit compact form. Additional instructions would be required to unpack the data and make it usable by the pre-wired 32- or 64-bit data paths existing in general purpose processors. The circuit on the right would not require any additional changes.



KEY TECHNOLOGICAL CHALLENGES

- Building Blocks Inadequate
- Tough To Program
- Minimal Runtime Support
- Verification
- Innovative Algorithms

11

Blocks

- Create configurable components at various granularities

Program

- Automate hw/sw co-design
- Develop high-level programming environments
- Reduce compilation times

Runtime

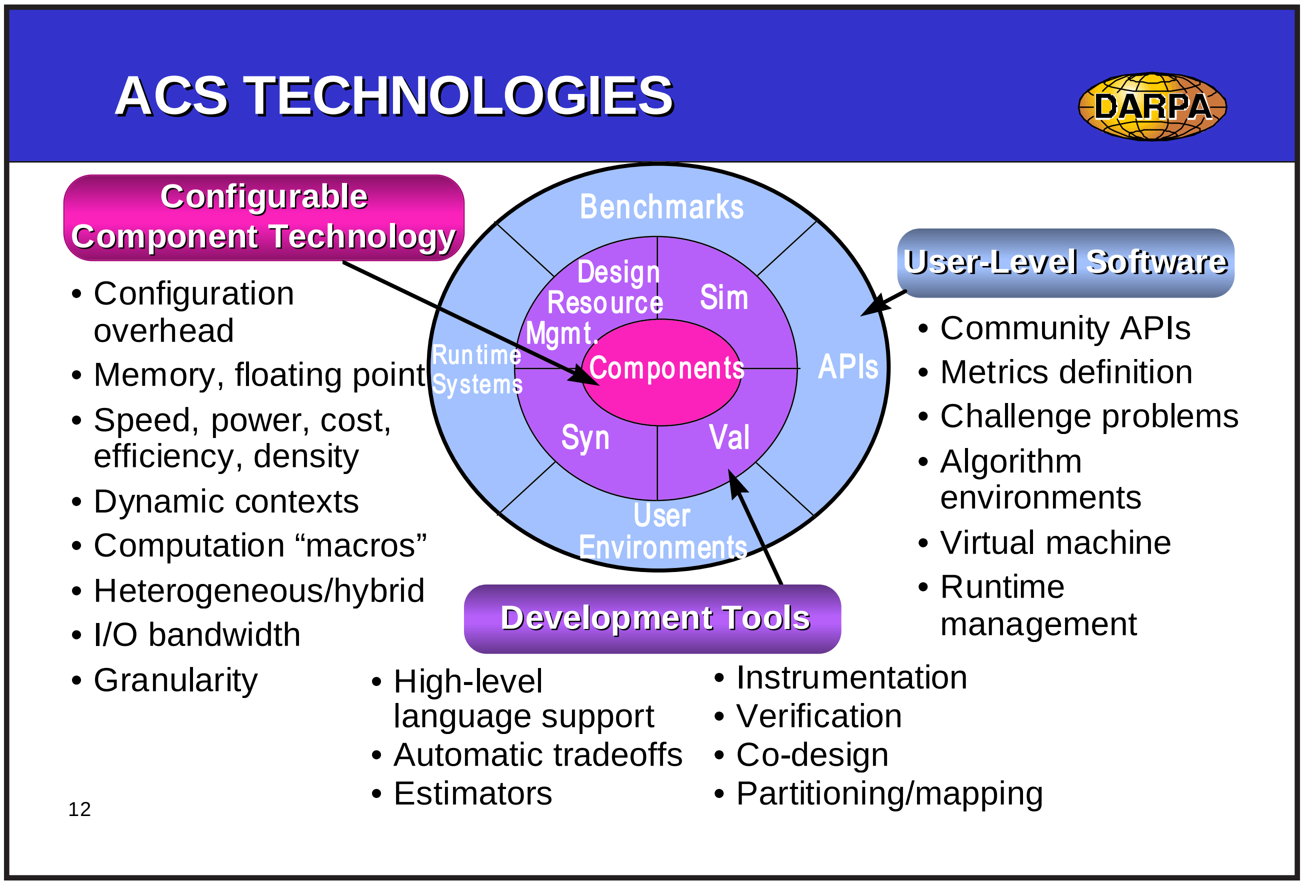
- Automate runtime reconfiguration down to the datapath level
- Support dynamic logic and datapath level reconfiguration
- Explore/exploit gate level fault-tolerance in the technology

Verification

- Develop techniques to rapidly verify that each dynamic mapping will perform its intended function
- Develop techniques to ensure chip integrity

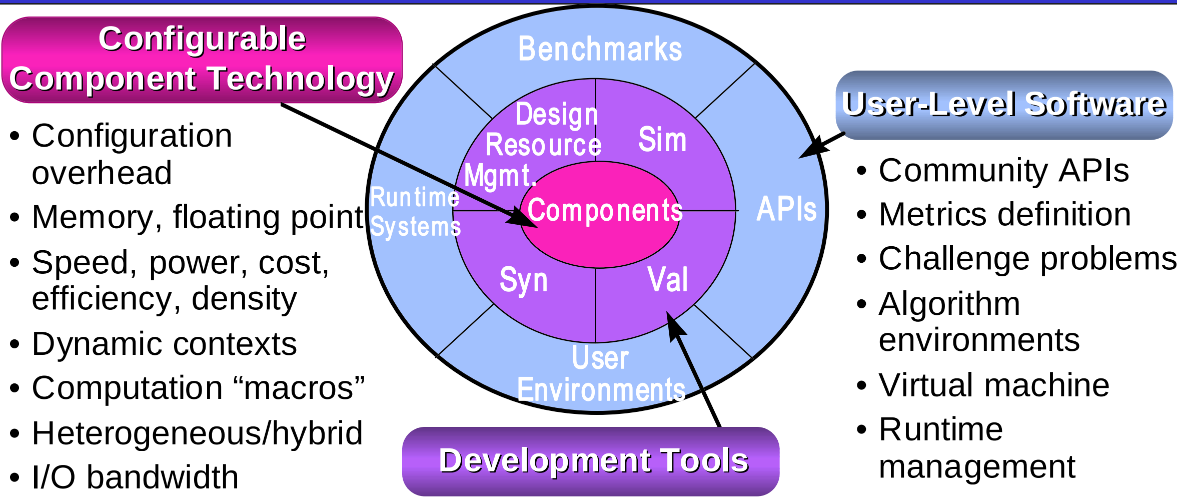
New Algorithms

- Leverage ACS technology through new algorithm approaches (e.g., variable precision arithmetic)



ACS TECHNOLOGIES

DARPA



- Configuration overhead
- Memory, floating point
- Speed, power, cost, efficiency, density
- Dynamic contexts
- Computation “macros”
- Heterogeneous/hybrid
- I/O bandwidth
- Granularity

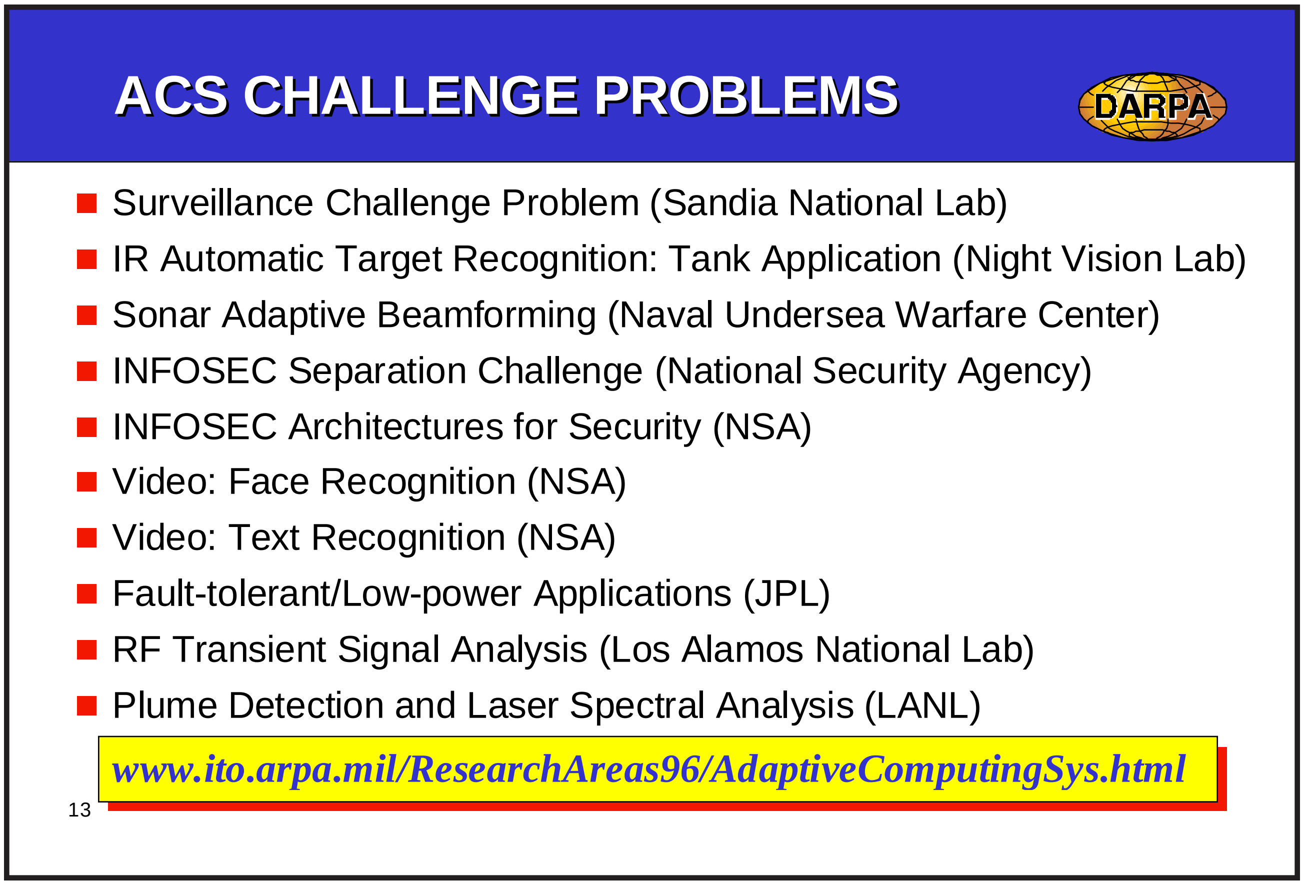
- Community APIs
- Metrics definition
- Challenge problems
- Algorithm environments
- Virtual machine
- Runtime management

- High-level language support
- Automatic tradeoffs
- Estimators

- Instrumentation
- Verification
- Co-design
- Partitioning/mapping

12

Various elements come into play when creating the requisite technologies necessary to make ACS viable. This “onion skin” diagram shows on the outer ring those elements seen by the user of ACS: the User environments, APIs and runtime tools necessary for him to formulate his application. The center ring shows the development tools that might be used by the ACS vendors: compilation tools, partitioning and mapping tools, estimators that could be used to provide feedback to the user indicating how many gates will be utilized, simulation and verification tools, etc. The core shows the actual hardware components that are an integral part of the program and enable reconfiguration. Here we find: gates, memories, computation macros, heterogeneous architectures consisting of FPGAs, GPs and DSPs, IO support, support for dynamic contexts, floating point operations, etc.



ACS CHALLENGE PROBLEMS

DARPA

- Surveillance Challenge Problem (Sandia National Lab)
- IR Automatic Target Recognition: Tank Application (Night Vision Lab)
- Sonar Adaptive Beamforming (Naval Undersea Warfare Center)
- INFOSEC Separation Challenge (National Security Agency)
- INFOSEC Architectures for Security (NSA)
- Video: Face Recognition (NSA)
- Video: Text Recognition (NSA)
- Fault-tolerant/Low-power Applications (JPL)
- RF Transient Signal Analysis (Los Alamos National Lab)
- Plume Detection and Laser Spectral Analysis (LANL)

www.ito.arpa.mil/ResearchAreas96/AdaptiveComputingSys.html

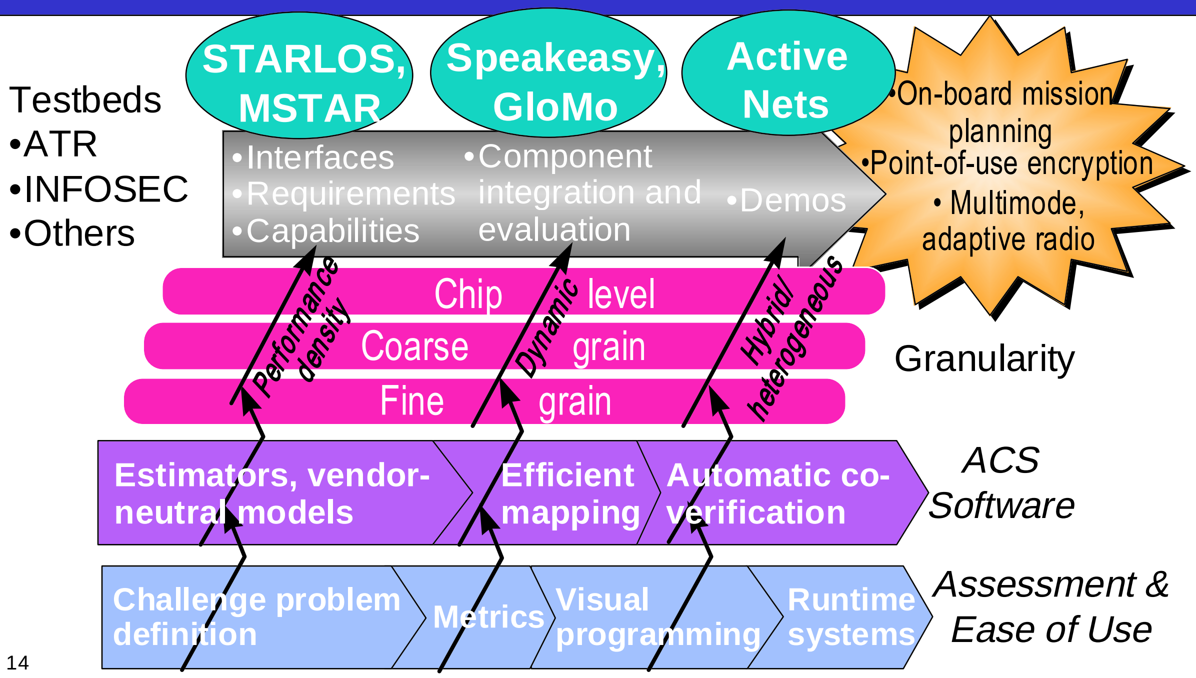
13

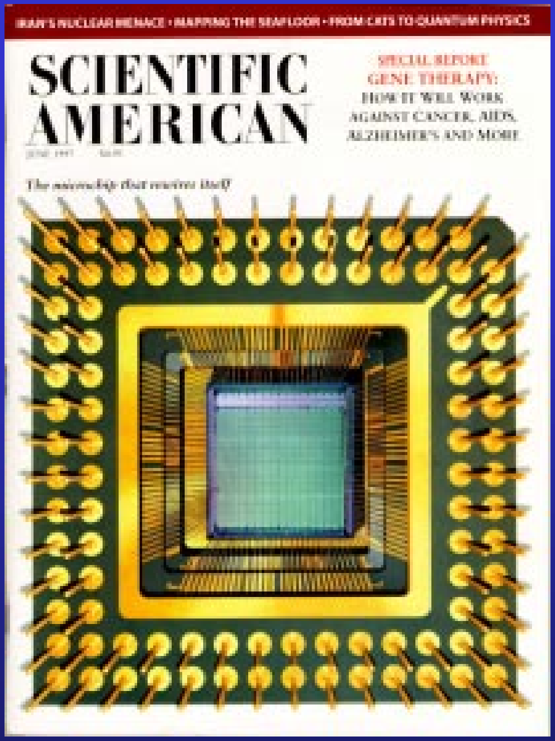
In order to provide a focus to the ACS community a set of military specific “challenge problems” has been identified. These problems are being presented to the ACS community, which is invited to formulate solutions using this technology.

An example is The Surveillance Challenge Problem provided by Sandia National Lab. It represents an automatic target recognition problem, where ACS will attempt to provide a solution within a cubic foot…a 500X size reduction over conventional approaches.

Details concerning the challenge problems are presented on the World Wide Web at the following URL address: http://www.ito.arpa.mil/ResearchAreas96/AdaptiveComputingSys.html.

ROADMAP


STARLOS, MSTAR
Speakeasy, GloMo
Active Nets
Testbeds
•ATR
•INFOSEC
•Others
•Interfaces
•Requirements
•Capabilities
•Component integration and evaluation
•Demos
•On-board mission planning
•Point-of-use encryption
• Multimode, adaptive radio
Chip level
Coarse grain
Fine grain
Granularity
Performance density
Dynamic
Hybrid/ heterogeneous
Estimators, vendor-neutral models
Efficient mapping
Automatic co-verification
ACS Software
Challenge problem definition
Metrics
Visual programming
Runtime systems
Assessment & Ease of Use


IRAN'S NUCLEAR MENACE • MAPPING THE SEAFLOOR • FROM CATS TO QUANTUM PHYSICS
SCIENTIFIC AMERICAN
SPECIAL REPORT
GENE THERAPY:
HOW IT WILL WORK AGAINST CANCER, AIDS, ALZHEIMER'S AND MORE
JUNE 1997
The microchip that rewires itself

15

The June 1997 issue of *Scientific American* has a feature article, “Configurable Computing,” devoted to this topic. It was written by two principal investigators in the DARPA/ACS program, from UCLA, and provides an excellent background in the area of configurable computing and how it might be applied to solve many of the challenging problems presented here.